CARLO QUINONEZ: With any new technology, people always over emphasize the similarities of existing technologies and marginalize everything that makes that technology different than the existing technologies. LEDs make color, therefore, they can replace light bulbs. Today, with 3D printers we have, hey, they print plastic. They can replace injection molding machines.

Yeah, doesn't quite work that way. But that's the only point of reference people have to these new technologies is what exists today. And that's the way, at that point, when I first started using 3D printing, that's the way I looked at 3D printing. The things that were different about it, like that it really can't replace injection molding, is what made it suck compared to normal injection molding.

So I definitely started off on that path with 3D printing, one I think most engineers do. They see it as full of promise, but realistically, it's a tool that doesn't do what you want it to do. Honestly, I think that problem's more with engineer than with the technology. So little about the project.

After the engineering, I actually got married after this engineering company started winding down. And so I just took a role back in lab, that safe, comfortable role of working on a science project developing custom hardware. In this particular project-- don't hit this button-- we needed to use a microscope to collect data.

So there's an objective. This is a microscope. This is an inverted microscope. Just so you know. The objectives are below the stage and this is the light source. So we're doing single cell imaging. It was a cell biology lab, and we needed to study thousands and thousands of cells.

And that's when I realized I didn't really want to be the guy in the lab for thousands and thousands of cells, and so you do what an engineer does. You build a tool that let's somebody else do it for you. And so that's where I started looking at, hey, I know 3D printing, I know CAD, I can do great on this.

You have the stage including the desk holder. And then you need a sample holder, because samples usually aren't big rectangles like that. So this was a plate for the standard 35 millimeter culture dish. And you turn on the objectives, there you go. You can imaging your samples, start collecting data.

We're just going to get rid of the condenser for now. But, of course, this is great if you have dead cells, because normal living cells don't do well at room temperature in exposed atmosphere. So started working on a cell culture incubator that fit on top of microscopes.

Now, these are commercially available products, but, like I said, you usually need to do a lot of customization to fit the right sensors you need or whatever quirk, idiosyncrasy in your research. There's always customization involved. This is where I thought, hey, 3D printing could be great.

It's a small object. It's about the size of a paperback book. Flat, low profile. Totally seems like a nicely, well-suited project for 3D printing. And the first version I built looked a whole lot like commercially available units.

Usually commercially able units look very similar to this. Maybe I've rounded corners or something, but it's largely the same. I actually just went through data sheets of commercially available units and made something 3D printing that's comparable. And that's what kind of got me started on this really interesting pad I've been taking on 3D printing.

So that was the first of many incubators I ended up building. This was back in 2010. And by the way, if you guys have questions, please interrupt, stop. I'd rather make it more of a conversation to make sure this is working for you guys. That's the right button.

Like some of the things I'll plant, novel features, right? Now the connectors are on the back. So the experimenter sits over on this side, and so we want all the wires running off the backs. That way, people can reach into here without having to worry about knocking it around.

The sample chamber wasn't just big enough to hold one sample dish. It could hold a variety. And actually, the experiment we're using needed microfluidics, which is a way of automating food handling operations. And the other thing you need for growing cells is to keep them alive in high humidity.

So this is where we're getting to some interesting pictures with 3D printing. So the gas line's in the back, but for whatever reason, I put the humidification area in the front. And so basically what happens is the gas comes into this end, moves along, snakes down this path, and this water is heated the same as the cells, so it picks up some humidity before being just fed over in this little gap right here in this [? culture ?] area.

And it worked. It did the job. But it was not ideal. It took a long time to recover humidity. If you

take the cover off, and put the cover back on, it took a long time actually to recover humidity. Another thing that's notable about this is that we used aluminum.

A ton of aluminum. Not a ton, but a lot of aluminum. A thick aluminum face plate, because aluminum provides high thermal conductivity, which is important to ensure even heating. By the way, all these details, they will make total sense why I'm going into them in just a minute. Or maybe 20 minutes.

High thermal conductivity, that's how come we can put two heaters on here, and get a pretty even heating across the whole surface, This is like one of the fundamental assumptions in building any oven is that you're going to have a thermally conductive chamber. You can put a flame underneath it, and spread the heat around for you. Does all the hard work.

And then two tinfoil heaters so they were on the bottom. And if at some point, I needed to rebuild it, 3D printing. Got to love it. It's easy to do it. We had to, at that point, I think the campus had just gotten a 3D printer, an Objet, a small one, and were just cranking away using it.

Second version. Actually about six months after I built the first version I stopped and asked myself, why did I build a box? I have a 3D printer. I can make any shape I want, more or less. Why'd I start building a box?

And that was the first moment I realized I bring a lot of baggage to this design process. Preconceived ideas of how things should look. Everything should look like a box, because boxes are the cheapest shape to make. That's why we live in boxes.

But boxes are not natural. It's just a contrivance for our convenience. And we bring this baggage to every project we start attacking. So I started forcing myself, hey, start avoiding the box.

Some shapes like the bottom, the aluminum plate is [? CNC, ?] so didn't want to get too fancy with that one. But the top part, where all the magic happens, definitely started getting significantly more complex.

So now we started including a 3D print attachment points for little fans to improve temperature homogeneity. We added a lot more O-rings. Actually, the first design had no O-rings in the ceiling. This now design we had to build in the grooves, do all the work to make sure

everything's fitted right. But we added the O-rings to now provide a sealed chamber.

That lets us now use custom gas compositions to grow cells in. Because actually many cells need to grow in what they call blood gas, which is 5% CO2 in the balance air. But they need higher CO2 than normal in order to grow and be happy. And also I changed the design of this humidification area, because the other one just seemed not an efficient use of space.

And so the other things started working on is we wanted to keep this, because everything else was sealed and airtight. We wanted to make sure the gas entry into that side of the chamber where the cells are growing was also airtight. And so we had this mechanical interface.

These two little arms that stick out that drop into holes on the base, on the rest of this part, and that's how air comes into the nozzle here, goes around this path, and up through this green. That's the negative space that I carved out in the part. And so this is how the air gets into the chamber up in that corner.

And then when the air comes out, we capture it in this little hole, feed it back through these arms, and out to here. And because the whole chamber's sealed, we can capture that effluent gas, run samples, process it, whatever the case might be. This is where I thought it was kind of interesting, 3D printer, what else can we do with it?

We designed and built-- incorporated-- a rotameter with a variable area flow meter, flow rate meter. So that we could-- and you'll see the design of this-- it's just a tapered or tapering down cylinder. Put a little ball in there. With no gas flowing, the ball drops onto the bottom. With a lot of gas flowing, it flows up to the top.

Hey. Those are one of the oldest flow meters ever. The ones that you buy commercially are calibrated. For this we just wanted to make sure, hey, is the gas turned on? And so we just had little rules. It has to be between these two notches. Or as long as it's in there, we're good. We didn't need to get more precise than that.

But this is kind of neat. There's some neat functionality you can do with 3D printers. This little tapered shape is hard to make, but it's easy with 3D printing. This got me excited about the possibilities of what to do with this complexity though, what to do with all that free complexity. And I really dove in hard on the next redesign a year later.

Now, this version carried over a lot of the same concepts. We started the aluminum base plates, but now we chopped it up into two different temperature zones. There's now this gap

here you see, so there's two pieces of aluminum there.

And this now has three active temperature sensors and two passive temperature sensors. So now we're actually reading out data from five different points on this little box that's the size of a paperback book. And it was so sensitive at this point that I could tell you whether the door of the room was open or closed.

I logged the data over a week time. Normally, I'd only look at it for a couple of hours. Yeah, temperature looks great. Once I logged it out by accident, because it took a really long time, like a week's worth of data, and that's why I noticed these really long transients. Step functions in the temperature data that clearly were not normal. And it was a door.

I had to put a web camera in the room to figure that one out. But that's when I was like, hey, we can actually build pretty high precision instruments, sensitive ones. This is run by an Arduino and everything is available. There's no custom. Technology's all off the shelf.

Again, we just followed the same server fans. The water chamber's in the back. Two heat zones now. This zone is independent now, so we could keep the chamber temperature independent of how much heat we're applying to this back zone where all the water's being vaporized to humidify this chamber. That gives a lot more dynamic control over the ability to humidify those chambers.

This is where the project ended after three years. We used it to collect some data. Published some papers. But it definitely stuck with me as a project. I went to work for a couple of years at another company. When I left that company, in the time between companies I decided to pick up this project again and explore that question.

See, I assumed I needed metal in making an incubator. That was an assumption on my part clearly. Test it. Do I really need that? Can I build an oven out of plastic? And this is basically just a proof of concept test part.

The way it works is it'll blow our fan here. These are big flat, they're actually power resistors. But they're 20-watt power resistors, so they would get nice and hot. They're nice big, basically thin film heating elements. Little blower.

And what's not really clear from this is that there is one channel that feeds this branching structure that's 3D printed. This front part has this very complicated branching structure more

like the alveoli in the bronchi in your lung. But this is what gets that heated air and distributes it evenly over that entire surface.

Because keep in mind that plastic's essentially an insulator. It does not conduct heat very fast at all. So this was a period to learn about how to design self-supporting structures. Really get to the limits where 3D printing is, understanding its quirks and idiosyncrasies.

We revised this design. It took a long time to heat up, so we minimized the thickness of this wall to make it more responsive so we can do temperature step change operations easier. And this worked as a proof of concept.

So I managed. This is when I started working at FATHOM now. And what I was brought on at FATHOM to do was to do some internal and external research projects. Internal with some process development stuff like how do we use 3D printing in our shop more efficiently like the intersection of conventional manufacturing and additive.

And we also did work for clients or companies that are interested and curious about 3D printing would hire us. And so I talked my boss into letting me, hey, take this to the next version, to the next level, as a research project that we could publicly talk about and showcase. That got us to Project Pyra.

This is 14 inches square [? of raw ?] eight inches of usable space on the inside. It's an oven. Entirely plastic. There's spinning fans over those heating elements. I have some stills of this. And it works.

Inside the base, there's about 300 watts of these thin film heaters. And we're heating it to over 200 degrees Celsius. We can make a reflow oven out of this basically. And the plastic we used was ULTEM.

It's one of the Stratasys'. I think they're a huge competitor to Vantage, a variety of other 3D printer. They can print in polyetherimides. The one we used was actually rated for food contact up to 180, 190 Celsius. That's like 375-400 Fahrenheit. Hot enough to bake with.

The design of the top included the self-supporting. This was all done on an FDM printer, of course. So these walls were all self-supporting. Heat was distributed, the heated air, through this big central channel, and into these little tributaries, and drops it down into these 3D printed heat exchange elements that are incorporated into each one of the, pretty much the entire surface.

Had to figure out how to do this two different ways, because the structures that work for the sloping side. And that's why it's a pyramid too. 45 degrees are the easiest angle to print with an FDM printer. Horizontal surfaces, challenge. Sloping sides, not a problem.

But if this was still just an exploration project to understand what are the kinds of approaches. Because when you start building trees like feature trees that are as long as Pyra. Does anyone want to guess how many features in the feature tree, the history, were in to make this final part?

Keep in mind there's only, what, four pieces. Each one is a different color. There's a blue piece, green piece, yellow piece, and the fan. Four total for the whole thing.

AUDIENCE: [INAUDIBLE] .

CARLO QUINONEZ: 1500. Yeah. It's really hard to design a robust CAD file where you can go back and make changes to it without it blowing up. Because we've all had that experience, I'm sure, right? We get some CAD. The guy was in a hurry, didn't re-label stuff, didn't clean it up.

And now you touch anything it just blows up. It's faster just to rebuild it from scratch. I rebuilt this feature tree like three times. At some points, this is where I understand very well the limits of fusion at least where I was two years ago.

AUDIENCE: So it's a [INAUDIBLE].

CARLO QUINONEZ: Oh, yeah. It's 100% hand. I say it was optimized by me. I did this. This was 100% purely a passion project. There's no way any reasonably sane engineer would do this.

To sit down for like a month and a half, trying to do four parts that have 1,000 features in them. You'd walk away from it, right? Couldn't be done. Whatever you'd say, you wouldn't stick it out.

I would really want-- I always say that 3D printing is like a relationship. If all you're going to do is complain about it, it's never going to work. You have to embrace it for all its idiosyncrasies and all the shit pisses you off about working with 3D printing. Just got to figure it out.

And this was about. What's the thing you get for free with 3D printing, or one of the things you get for free we always say? Features complexity. It costs as much to print a complex crazy part as it does a really simple block.

The cost of 3D printing is driven by the amount of material. And that's pretty much it. Geometric complexity has very little role in the cost of parts. Actually, it has some, but it's not a main driver for it.

And so this was neat. I definitely learned a lot about how to structure patterns, how to build the trees. I figured this was all on the job training. So that's where we ended last year. Out of that, I got a lot of I think good perspective on this. I, again, get attracted toward complexity. Like there was gold in those hills.

And this is something we've all struggled with as a service bureau. How do you judge a part's complexity? Because the more complex a part is the riskier it is to 3D out, because you got to do support removal operations, whatever it is. It's more delicate if it has a bunch of complex features on it. We've always looked for algorithms that we could use to quantify part complexity.

And there's three that we found. It's not well, yeah. I'd say it's not a well [? forward ?] field. It's tough. You have Spies Ratio, which is a pretty common one. We've been looking at Spies Ratio for coding algorithms. Is this the ratio of the volume to the volume of the bounding box essentially.

So what percentage of the bounding box does the part fill? That goes from 0 to 1. Part-Complexity Evaluation Model. Mouth-full, but they didn't come up with a shorter name for it. This was originally used to judge the complexity of STLs.

And it's just a metric measure based on the number of facets that STL has, and then corrected for the size. So the same number or facets in a bigger STL file is less complex than if you made that file really tiny. So there's some correction there for the size of the part.

You know I just realized. These two are switched. Sorry. And the last one that we found was interesting was this mean connectivity value. And so that requires voxelizing a part. Turning it into a voxel-based part, and then counting how many other voxels each box on the surface can see into the part.

So you're a box on the surface. You look back into the part. How many voxels can you see? So if you're a sphere, you can see all the boxes in the part are visible to every voxel on the surface if it's a spherical object. If it's some other shape, it's not, it's going to go down. The

more like a lattice network, you see very little of the cells.

But that looked interesting except that required a lot of programming to do. And so we've been focusing on the middle ground literally, this part-complexity evaluation model. Because it has more data than just the ratio of bounding box to the part volume. Yeah.

So just to give you guys some external validation on these metrics and why it's valuable, you guys know the GE maintenance bracket design challenge last year. So this bracket is used to hold the engine during maintenance operations. Not a critical flight safety issue. And currently, they attach those brackets to the engine, so that way they can hoist them off the airplane, the engines when they need to service them.

So the bracket's only ever used when the plane's on the ground. They're currently machined out of titanium bar stock and it weighs four pounds. And so based on industry figures in 2012, one pound of weight to an airplane adds about $100 in cost to an average trip.

So if they can shave a couple of pounds off this multiplied by number of planes and number of trips, you're talking about saving up to $10 billion a year industry wide by shaving some pounds off this part. And this is the context of why they started this. Is anyone not familiar with this engine bracket design challenge?

All right. So it was started in 20-- what the heck-- '15. I think it opened up in the early part of the year, and it closed later in that year. It was an open sourced competition. Anybody in the world could submitted design to GrabCAD. GrabCAD was the sponsor, running, coordinating the contest.

The restrictions were minimal. It's going to be 3D printed in titanium. The minimum feature size is going to be 50 thousandths of an inch. And the top 10 designs as judged by this set of rules that they had would each get $20,000 in prizes. They got over 700 entries, which is not as many as I would have thought for small brackets like that big.

Redesign it. It opened in June 2013 and they closed in November 2013. And there were over 700 entries. Right This is the last place finalist. Weighed 424 grams, and it's part com-- oh. I guess maybe I should start with this one.

The original design, the machine from solid bar stock. Two kilos. And a 68 of that part-complexity evaluation metric right. Last place finisher. 424 grams. Already shaved like 1.5 kilos off. 533 was its part-complexity measure.

422 for the next finisher, and 500 part-complexity. You'll see as the grams are going down, complexity keeps going up. 499. 403. I just grabbed these CAD files off of GrabCAD.

And this is based off of the number of facets, number of faces in the B-rep, in the step file. Not based on STL facets, but the number of B-rep faces. I look at this. This is a part that an engineer is making completely by hand. It's 100% to skill.

For some engineers, I'm sure they can do better, more efficient designs with fewer surfaces. But whatever. To see what this looks like. We can plot it out. The higher ranked you were, at least in the finalists, probably all the finalists were all equally skilled engineers, highly skilled engineers in the finalists.

There's one other outlier too I didn't put in here just because it looked like it was done by a game designer. It was very complicated, but it finished seventh. And so I was like, hey, this is good. This is actually somewhat validates this hypothesis that I've had that complexity can be useful.

This next part of the talk is some of these really concrete real world examples of how we've harnessed complexity. So I want to stop and define what the word harness means. Dictionary definition means to control and make use of a resource, especially to do work. Can we harness complexity to do work?

These are where now these details come back. Poor thermal conductivity. Normally, engineers would rely on the simpler design. You rely on the passive thermal conductivity and inherent property of the aluminum or metal to spread the heat around. Why would you spend any of your effort on it?

Leading to designs like this. I have a complexity of 580 for this efficient distributor heat compared to 32,000 for the complexity in this part in just the head. So this is I think a good case for how you can use complexity to address a real world problem that you're normally addressing somewhere else. Everything's tradeoffs in engineering.

So yeah, the aluminum thermal tub in the 2010 design has a thermal conductivity of 200 watts per meter Calvin versus ABS that we're using in this. Thermal connectivity of 0.2. So 1,000 times lower than aluminum. Complexity went up by a little less than 1,000 times.

But this is starting to get a feel of my mind of where this is going. This is a design I'll talk about

in just a moment. But this is an incubator for another project that we're working on.

Here we're using heating fluid-- cooling fluid-- to cool down a sample chamber. Just like the thermalators, they have an injection molding machine. We run through a silicon oil into this. It circulates around, and cools down the chamber.

That is an example of a heating fluid that has a high specific gravity and a high specific heat. So it's pretty easy to heat and cool stuff down with oils. With air, not so easy. Very low specific gravity. It's thermal specific heat's not too different, but it's not very dense at all.

So it's going to be harder to heat and cool with air than with an oil. And just like you'd expect, the complexity of designs differ dramatically. With the oil, easier material to work with, lower complex design than with the air.

The other thing Crucible had was-- maybe I'll back up a little bit. Talk a little bit more about Crucible. So Crucible is this far-out there research project that I managed to get. So Crucible's a cell culture incubator for growing living cells in conditions that simulate Mars.

Near vacuum. Near cryogenic, as low as minus 70 Celsius. Custom gas atmospheres. Custom controllable lighting. It's an environmental chamber for simulating the environment on Mars. Large of it's all 3D printing. There's no machine parts in there at all.

AUDIENCE: About how large is it?

CARLO QUINONEZ: It doesn't fit inside a 3D printer. You can put one dish in there basically. The cost of 3D printing went out on a Fortus in ULTEM is like $2,000. You can't buy them off the shelf. Nobody makes it. It's all bespoke engineered stuff.

They're like two of those things exist in the world today. One's in Cape Cavarnel, the other one's somewhere in Europe. There's only people who can simulate conditions of Mars and try and grow stuff on it. Crucible costs $2,000 versus $200,000 for the engineered conventional one.

I'm sure this one took a lot more design work and experience to get there and build it. But one of the challenges besides just the heating and cooling of it, how do we heat and cool the chamber? Oil versus air. We had to deal with, I mean, this case we took the easy route-- the oil.

But the next challenge then was insulation. And minus 70 you're going to need to have significant amount of insulation around it. So here we spent time exploring different infill patterns and evaluating infill strategies for their insulative properties, not just their ability to cut weight and be mechanically strong. Because as engineers we also design stuff that stays a certain temperature because that also protects it from the environment.

And so here the experiments we're doing are pretty rough, but different infill strategies. The blue line is foam insulation-- the standard. And other lines are all different infill strategies. And we found the best for this.

And this is where more limitations of CAD's software starts becoming very evident. Because if you wanted to fill a part with a bunch of hexagons, not tall, long hexagons because that not very good insulation, but cells. You're never going to do that in CAD software to model in all of these little cells.

Because it's still closed cell insulation except we're manually building all the cells rather than just using a gas the way normally expanded foam insulation is done. This ends up being simpler to design in the CAD software, but took significantly longer in post-processing to have it generate the G-code. That's not G-code for Stratasys, but the G-code to run the printer.

Because out of the CAD software comes the solid body. Solid body has walls four inches thick. But then we have to go into that file and manually edit it to get it to make a bunch of closed cells in there rather than our [? normal fill ?] strategies.

And so what we found works best, at least in this case, was a hexagonal pattern, two to three millimeters big. And then every 12 layers, you put a full layer of solid raster infill. It sort of creates closed cells, but it definitely is an improvement over normal infill and insulating properties you get in infill.

The other challenge we had was airtight. We had people simulate 1% the atmospheric pressure of earth in this 3D printed part. And so people generally think it's impossible to print airtight generally. There's always exceptions, but generally, it's impossible to do airtight parts on [? FGAM ?] printers. And I'd say how we qualify that, I would qualify it is if you're using a normal slicer.

Slicers are pieces of software that translate a B-rep into a series of G-codes where you're optimizing for printing speed and part weight. They're not optimizing for anything else. And if

you're using software that is designed to minimize weight and material usage, why would you expect that it would make airtight parts?

It had to be this. It was my hunch going into the project, because plastic. Plastic's airtight. There's no reason why a plastic printer shouldn't be able to make airtight parts. And it all came down to the slicing strategies and the G-code paths and the tool paths.

We ended up having to develop a custom slicer. We wrote our own slicer inside of Fusion using the API. It was definitely slow until we turned off the history. It would take as long as 45 minutes to slice one file, and these weren't particularly complex files. It's just like a tapered surface four inches thick.

But these are examples of the extrusions that it produces. And that gets then fed into a translator where each surface is interpreted as a tool path. Oh, and the different strategies we used for generating these resulted in different amounts of leakage. So the green is a flat piece of acrylic.

Perfect sealer. No pressure loss over time. The other lines represent different strategies to generate airtight parts. And so some of the stuff we looked at was how to stagger contours on adjacent layers. When to do overfilling, so put in more plastic than is necessary, and just spluge it all out.

And there's definitely some variations there. And we've essentially written the beginnings of a general purpose slicer-- well, not general purpose-- slicer for producing generally airtight parts on your FDM printer. So that's essentially the bulk of the new stuff. I do want to talk a little bit.

This is Crucible. It's working. The next step is to finish it. Work out all the bugs, and get it out as open source software. I mean, open source hardware design.

I think even though this is an external client project-- it's a paid project-- I still think it's going to be a great example to show how you can effectively and efficiently utilize that free complexity you get with 3D printing to generate some parts that will hopefully inspire and certainly make it clear to engineers that haven't really dove into 3D printing that there's a lot more to it than just printing out plastic. It's fundamentally a different design approach.

And so the next step is going to be releasing all the designs of the software and hardware. It's been written as open source. And this is why I think academia is a great place for it. They already have a culture of sharing, they're all technically sophisticated, they all have 3D

printers, and all their needs are sufficiently different that commercial solutions don't provide good solutions.

[? No ?] [? very ?] good answers for them, and they're already used to doing customization. The applications are low value, low volume, high value. Everything that makes it perfect for 3D printing in an open source lightweight innovation approach.

The biggest one-- I should have put that quote in there-- there's a great quote by the CTO of Intel. The quote's, "Technologies have regressed to such a point that what we can make is limited by our imaginations." Seems uplifting. Until you stop and think about it that we're the weak link in that chain.

Our imaginations are the ones that suck, not the technology. [? People ?] [? with ?] 3D printing are always waiting for a better printer, better materials, better simulations, something before it'll take off when the limiting factor is us-- the engineers-- who'd never rationally build parts with a thousand features in it.

I think something fundamental does have to change in that mindset of engineers. KISS. First thing you ever learned. What's it stand for?

AUDIENCE: Keep it simple, stupid.

CARLO QUINONEZ: If you've been beating at engineers their entire life keeping stuff simple why would you expect them to be able to effectively use a 3D printer? Telling them never to use complexity, and if you give them a tool who's really only value-- one of those two big values is complexity.

Because let's be honest. 3D printing surface finish is worse than chemical manufacturing. It's more expensive than commercial manufacturing. Lower processivity than conventional manufacturing. Poorer material properties than conventional.

In every other way you can measure it, it kind of sucks compared to normal manufacturing. Its advantages lie in no change costs and free complexity. But we're telling engineers keep things simple, so they're never going to take advantage of the complexity. And then the change costs, have you guys used PLM software?

How easy is it to make changes? [? Provo ?] workflows. An entire engineering culture, [? programming ?] culture, is run eliminating change. You lock it down, and you don't change your design anymore. So KISS, keeping stuff simple and not changing it. And that's the two

strengths of 3D printing right there.

So I've been trying to get this out. CAP-- complex is practical. If you have an empty spot in your part that's relatively simple, make it more complex. That's what I ended up doing with that rotameter, the flow meter. Had a nice simple flat front to it, I was like, I can do something here. What can I put there?

What can you put in the infill to make it better? The way I look at it is complexity is like the physical embodiment of your talent as an engineer. You spend more time crafting your part, like the gem cutter, spends maybe an hour before deciding where he's going to cut that gem.

Engineers are essentially doing the same thing with CAD software and vertical surfaces. We add surfaces into parts where ever we think we need them. But we need to learn how to run towards complexity, which is weird. It's like running toward the harder solution. In the end, that journey is worth it with 3D printing.

We're going to end it. This is the highest part complexity of a single part, those six incubators that we talked about, the six versions. In these designs, complexity has been going up about fivefold every year.

And I love this comparison of Moore's law, because Moore's law is the number of transistors per unit area will double every year. And let's be honest. Most transistors are not placed by humans. They're like a grid of memory cells. They all look exactly the same.

But the number of transistors has been doubling every two years, every year, I mean. If we're going to get to this future, and I really do want to get there, because who can imagine what a part that has a complexity of 100 million looks like? What will that part do for you? That's kind of exciting.

The challenge for some of us or people who write software is how you going to write software that supports that level of complexity? And that's one of the things we're working on is a build toolchain to make it easier for engineers to build more complex single parts. And a lot of those challenges mirror the challenges faced by software developers.

Software developers have built up tools of their own to deal with more complex software, paradigms, everything else. I really want that future. So that's where we're going to end it. This is my contact information. If you guys have any questions we've got time for questions.

AUDIENCE: You said you couldn't represent the closed cells inside that with CAD software?

CARLO QUINONEZ: Yeah. There's like 30,000 little closed hexagons. Little hexagons that are three millimeters across and like one millimeter tall. There's like 30,000 of them in there.

AUDIENCE: So you used your post to do that?

CARLO QUINONEZ: You used what? Yeah.

AUDIENCE: Your post process.

CARLO QUINONEZ: Exactly. But the challenge there is that the engineer who knows our value. We know we need our value 12 in this much space, we had to figure out how to capture that in the CAD information so the post-processer can get it.

And insulation is not isotropic necessarily. It may block better in one direction than in another direction. It's certainly going to be a case if there's little pancakes. Essentially what we're building. So that was all done in post. But that's the kind of stuff that really should be done in the CAD.

It's just CAD software today doesn't handle that. Try doing some pattern operations. Make a little [INAUDIBLE] cube. See how many of those you can pattern before your computer just stops working.

AUDIENCE: Can it get a little more complex [INAUDIBLE] like the first exam that shows that tree branch structure. You got to do it by hand. If you want to get much more complex and much more fine structure, [INAUDIBLE] final structures you have to have software because otherwise [INAUDIBLE].

CARLO QUINONEZ: Let me ask this question. How many software developers are there, by the way? One, two, none? Two. Three. Those guys are mechanical engineers. Perfect. So how many files do you use to build one part?

What's the relationship between number of files to parts? Come on mechanical guys.

AUDIENCE: One to one.

CARLO One to one. There's one file for each part in exactly one to one. [? Solid ?] works that way,

QUINONEZ: right? How many different files are in a typical software application?

AUDIENCE: One to several--

CARLO QUINONEZ: Thousands. I don't even know, because my computer does it all. It just sucks them all in and builds it. Tools like that like, even just fundamental things. This preconceived idea that it's one file, one part. That's not the case.

And this is the challenging part of new paradigms. You should question everything, but if you start questioning everything you'll get nothing done. And that's always the hardest part is what do you question and what do you leave for another day. But that's one that's probably whose time is coming up very shortly.

And that's what I love about Fusion. Because Fusion unlike Solidworks and most CAD software has no differentiation between assemblies and parts. They're all just designed. There's no penalties. There's some penalties, but it's its own thing. It's a little different than the way normally CAD software approaches the whole assembly.

But Fusion's what made this possible, by the way. There's no way this would have ever worked in traditional CAD software. You can do all your modeling operations at the part level. Once you get into the assembly level, though, you can only do a very, very limited subset of modeling operations.

So that discrepancy is a barrier to building more complex stuff, because you handle complexity by breaking in smaller chunks. That means multiple files or something. But you need to separate that stuff out, and that means we need to get away from one file, one part.

Cool. Thank you, guys. Please reach out and please fill out the speaker feedback form, because I would definitely would appreciate it. My first time talking and want to come back. Thanks.

[CLAPPING]

AUDIENCE: What's the print time on the--

CARLO QUINONEZ: The Crucible? It's like three days solid printing. But computer guys have a great answer to how do you make a slow process faster. Just put a bunch of machines in parallel. In the '80s, we used to think that what you needed to build was faster computers. Right around the '90s, it

was like, wait a second, we're just going to build cheap machines and put them in the same room.

AUDIENCE: Yeah, how big was it [INAUDIBLE]? I can't even imagine the number of features [INAUDIBLE]. How long was your [INAUDIBLE].

CARLO QUINONEZ: In Pyra, because I didn't have a good-- the oven, the pyramid shape-- I didn't really have a good sense of how to break it up into files. I didn't do a good job of it. Learned a lot.

That one I actually had to reboot the design three times. So 500 features in one file, and I was like, all right, I just have to exploit as a step, import the step, and then start building on the history on top of it. I had to do that twice. So I have three files each one holding one third of the history.

Now, with Crucible, what I did was use the design includes to start breaking up the complexity that way. But that has its own set of limitations too. Because especially now with the branching and merging and how it's implemented. The closest thing in software would be git submodules where you actually have to bring these other parts you're including into your repository and keep them in your version control system too. Which doesn't really make a lot of sense, which is why nobody uses git submodules.

AUDIENCE: I have something going [INAUDIBLE] because I just walked back there before I leave. I'm very interested in [? factors. ?] If you see my stuff that was 3D printed [? AD ?] arms. If you print something bigger then it's more complex but it's more [INAUDIBLE]. It doesn't have that [INAUDIBLE] detail. My idea was fragments. So I walked back there and I had this software that's called Dynamo. Basically, [? boxer ?] [? list. ?]

CARLO QUINONEZ: Yeah. It's generative design software for sure. I know AutoDesk also bought a voxel-based modeling tool too. It's kind of neat. I just want to know when they're going to bring it into Fusion or Inventor. Because the only guys that get to play with Dynamo are architects right now.

AUDIENCE: Yeah. I looked at the factor structure and saw [INAUDIBLE] was a little different than that, how is that factored. It was more like other systems. But then on the other side was Dynamo. And I could imagine a mixture of some [? factors ?] that rate attributes that I could have really large surfaces for.

CARLO QUINONEZ: Theoretically infinite surface. An infinite surface.

AUDIENCE: Depending on your [INAUDIBLE] size.

CARLO QUINONEZ: Yeah.

AUDIENCE: I teach something later. [INAUDIBLE].

CARLO QUINONEZ: Nice. Activate.

AUDIENCE: [INAUDIBLE]

AUDIENCE: Are you a Fusion 360 user?

AUDIENCE: Yeah.

AUDIENCE: Rule number one.

CARLO QUINONEZ: Activate.

AUDIENCE: You haven't been--

CARLO QUINONEZ: Oh, yeah. No. Dude. You get. This taught me. Pyra taught me the importance of activating the right shit.

AUDIENCE: It's not just activate. [INAUDIBLE] Rule number one is if you don't know what to do make a component and then activate. You only need a few components to activate. And then everything you did afterwards is stored in that component so when you export that component. You can't even complete a sentence.

You export that component, it exports complete design. A lot of users-- I teach that later-- a lot of users if you go by the Fusion 360 user interface, start with a sketch, extrude it, [INAUDIBLE] shade in the 3D geometry. Then you have your body. That's what a lot of users don't understand. There's a difference between body and form.

Then you have modify [INAUDIBLE] and then, some other time you figure out what kind of new

component you need. Then you take the body and make a component. [INAUDIBLE] because you're only exporting a [? dunbonnet. ?] Yeah.

CARLO QUINONEZ: Yeah. The [? dunbonnet. ?]

AUDIENCE: And it has other promises associated with it. And the question that somebody would ask why is that no different? [? Why would you make ?] first a component? Because in our workflows we didn't do that. Two or three other [? daisy ?] workflows where they wouldn't do that.

I started posting it because I kept posting the same thing to users. Then I make my own textbook. I use [? met ?] so I use nodes to make my own textbook. Copy, paste it. Somebody said, hey, after a hundred times posting, go put a sticky there. And now I put a sticky there and I've repeated that.

Other people started repeating the same thing just linked to that sticky note. It's so simple, but it's not so. There are other things in 360 that are [INAUDIBLE] that are so not obvious. When I go to the [? advisor ?] meetings usually a double click [INAUDIBLE]. How would you figure it out?

CARLO QUINONEZ: Yes. And clicking on a move operation and you select component. Clicking it in the graphics window versus clicking the name, different outcomes.

AUDIENCE: Yes. Because when you highlight the face and edge convert text of a 3D object, you're highlighting body. If you right click, you're moving the body. That's very dangerous. The effect of that is that they already looked at you design, 3D geometry there, your [INAUDIBLE] is there, and your sketch is there. That's [INAUDIBLE]. Because in Solidworks everybody works exclusively [? in viewport. ?] [INAUDIBLE] Click on something [INAUDIBLE].

CARLO QUINONEZ: So if I set my selection filter to component when I click on the viewport. I haven't tried that to see if that would work.

AUDIENCE: Select a component, you just double click on it.

CARLO QUINONEZ: You double click on it. Ah, there you go.

AUDIENCE: [INAUDIBLE].

CARLO QUINONEZ: Yeah. I'd forgotten that one.

AUDIENCE: Because how would you figure it out?

CARLO QUINONEZ: Yeah. I don't know. It's like on an iPad, how would you know to slide your fingers up toward the corners to make it go split keyboard. I don't know. You read a book? Who reads manuals?

AUDIENCE: Sometimes.

CARLO QUINONEZ: Sometimes.

AUDIENCE: I've asked, and said, hey, look at this! When I had my headphone plugged into an iPhone, I can [? keep this. ?] I can't [? make ?] [? folders ?] in that corner. [INAUDIBLE].

CARLO QUINONEZ: The other way you learn it is just by dealing with the fallout of not doing it right way. Doing these forced design challenges for myself, super valuable. You have to rebuild 1,000 features, you figured it out really fast. [? So ?] [? nuts. ?] Well, anyway. All right. Cool. Hey, nice seeing you, Peter.